TITLE:

2次元点渦系の粘性に関する考察 : 運動論的方程式での連続解と粒子解 (オイラー方程式の数理 : カルマン渦列と非定常渦運動100年)

AUTHOR(S):

八柳, 祐一; 羽鳥, 尹承

CITATION:

八柳, 祐一 ...[et al]. 2次元点渦系の粘性に関する考察 : 運動論的方程式での連続解と粒子解 (オイラー方程式の数理 : カルマン渦列と非定常渦運動100年). 数理解析研究所講究録 2012, 1776: 163-173

ISSUE DATE:

2012-02

URL:

http://hdl.handle.net/2433/171753

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録
第1776巻 2012年 163-173

2次元点渦系の粘性に関する考察
～運動論的方程式での連続解と粒子解～

八柳祐一
YUICHI YATSUYANAGI
静岡大学教育学部
FACULTY OF EDUCATION, SHIZUOKA UNIVERSITY

羽鳥尹承
TADATSUGU HATORI
核融合科学研究所
NATIONAL INSTITUTE FOR FUSION SCIENCE

1 初めに

計算格子を必要としないLagrange的計算手法として，点渦法がある [1]。点渦法は，2次元非圧縮非粘性流体方程式であるEuler方程式の「形式的」な解であることは，教科書レベルで知られている [2]。一方，点渦系にはその離散性に由来する拡散項が存在することを示唆する結果が，Onsagerに始まる絶対温度が負となる点渦系 [3,4] に関連する文脈で指摘され続けてきた。例えば，Joyceらは負温度点渦系の平衡分布を表す平均場方程式として，sinh-Poisson方程式と呼ばれるものを導出した [5]。sinh-Poisson方程式に合致する平衡分布は，本来，点渦系で時間発展させた分布を平均化したものであるべきだが，Matthaeusらは，減衰性2次元Navier-Stokes系で得られた平衡分布が，sinh-Poisson分布になることを示した [6]。また，負温度状態の平衡解については，Lundgren and Pointin [7]，Robert and Sommeria [8]，Eyink and Spohn [9] など，枚挙に暇がない。さらに，Leonardは，数値計算手法としての点渦法に関する概説において，"It now appears that using an increased number of point vortices of decreased strength will not yield a converged solution. ··· Ironically, best results with the point vortex method often are achieved by using only a few vortices with a diffusive time integration scheme." と指摘している [10]。また，点渦系は非平衡統計力学の対象として扱われることも多く，たとえば，Dubinは非中性プラズマ (案内中心プラズマ) の観点から [11]，ChavanisはBBGKY階層から点渦系の拡散係数を解析的に議論している。

これらの指摘を踏まえ，我々も粒子系である点渦系に拡散的効果が含まれないことに疑問を抱くに至った。点渦系においては，まさにボール同士が衝突するようなイメージでの衝突は起きないにしても，離散的分布ならば衝突に類するような効果が入っているほうが自然であると考えたわけである。本稿ではシミュレーション道具として一般化しつつあるGPUを手軽に利用する方法などにも触れながら，点渦系というモデルに内在する拡散効果の評価について，報告を行う。

2 点渦系

点渦系とは，Diracのデルタ関数の集合体として定義された離散渦群である。

$$\omega_z(\boldsymbol{r},t)=\sum_i^N \Omega_i\delta(\boldsymbol{r}-\boldsymbol{r}_i(t)) \tag{1}$$

$\omega_z(\boldsymbol{r},t)$ は渦度，Ω_i と $\boldsymbol{r}_i(t)=(x_i(t),y_i(t))$ は i 番目の点渦の強さ (循環) と位置ベクトル，N は全点渦数を表す。この点渦は，2次元非圧縮非粘性Euler方程式

$$\frac{\partial}{\partial t}\omega_z(\boldsymbol{r},t)+\boldsymbol{u}(\boldsymbol{r},t)\cdot\nabla\omega_z(\boldsymbol{r},t)=0 \tag{2}$$

の形式的解となる：

$$\begin{aligned}\frac{\partial}{\partial t}\omega_z(\boldsymbol{r},t) &= \frac{\partial}{\partial t}\left(\sum_i \Omega_i\delta(\boldsymbol{r}-\boldsymbol{r}_i(t))\right)\\ &= -\boldsymbol{u}(\boldsymbol{r},t)\cdot\nabla\sum_i \Omega_i\delta(\boldsymbol{r}-\boldsymbol{r}_i(t))\\ &= -\boldsymbol{u}(\boldsymbol{r},t)\cdot\nabla\omega_z(\boldsymbol{r},t)\end{aligned} \tag{3}$$

点渦系の温度を負にするためには，系の有界性が必要になるので，半径 R の円形境界を仮定する [4]。円形境界内での点渦の運動は，保存量の一つであるハミルトニアン

$$\begin{aligned}H &= -\frac{1}{4\pi}\sum_i^N\sum_{j\neq i}^N \Omega_i\Omega_j\ln|\boldsymbol{r}_i-\boldsymbol{r}_j|+\frac{1}{4\pi}\sum_i^N\sum_j^N \Omega_i\Omega_j\ln|\boldsymbol{r}_i-\bar{\boldsymbol{r}}_j|\\ &\quad -\frac{1}{4\pi}\sum_i^N\sum_j^N \Omega_i\Omega_j\ln\frac{R}{|\boldsymbol{r}_j|},\end{aligned} \tag{4}$$

を用いて表すと，

$$\Omega_i\frac{dx_i}{dt} = \frac{\partial H}{\partial y_i}, \tag{5}$$

$$\Omega_i\frac{dy_i}{dt} = -\frac{\partial H}{\partial x_i} \tag{6}$$

のように，ハミルトンの正準方程式風に表現することができる。

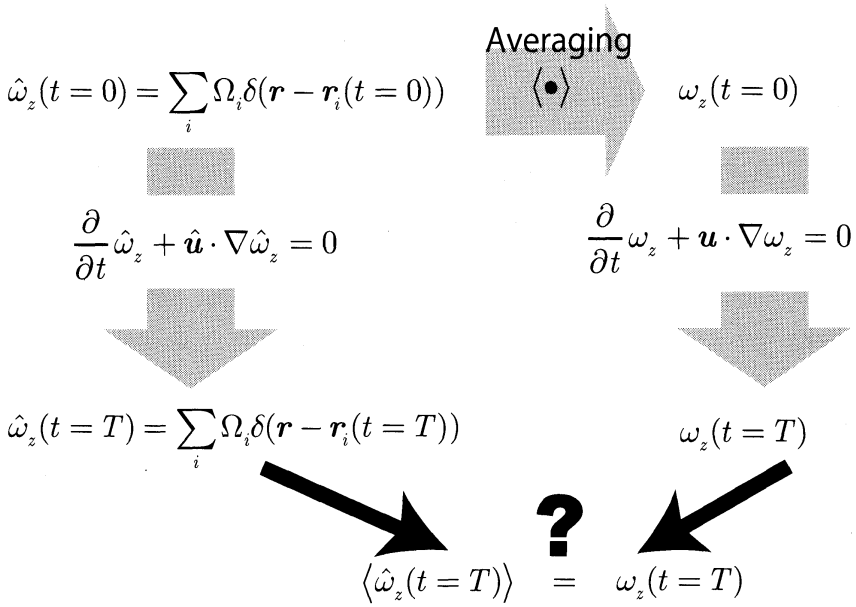

図 1: ミクロに追跡した結果を粗視化したものと，そもそもマクロなまま追跡した結果は，同じか？

3 ミクロな解とマクロな解

(2) 式の解について，もう少し考えてみよう。Euler 方程式はマクロな 2 次元流体現象を記述するマクロな方程式であり，本来，解は滑らかであるべきである。点渦系では，この滑らかな分布を点渦の分布密度で代替し，個々の点渦の運動を追跡して得られた分布を再度粗視化することによって，マクロな解を得る。一方，Euler 方程式は，もちろん，格子を用いたシミュレーションなどを行うことにより，マクロなまま解くこともできる。では，初期時刻 $t=0$ において設定した点渦分布を点渦法で時刻 $t=T$ まで時間発展させた結果と，初期の点渦分布を粗視化したマクロな分布をマクロなままオイラー方程式に従い時刻 $t=T$ まで時間発展させた結果は，同一の分布を与えるのだろうか，という疑問が沸いてくる (図 1)。この問題を考えるにあたり，プラズマ系での事例を参考にしたので，次にその事例の紹介を行う。

4 Plasma Kinetic Equation ～Klimontovich 方程式と Vlasov 方程式～

相空間中での粒子の位置を厳密に与えた相空間密度

$$\hat{f}(\boldsymbol{r},\boldsymbol{v},t)=\sum_i^N \delta(\boldsymbol{r}-\boldsymbol{r}_i(t))\delta(\boldsymbol{v}-\boldsymbol{v}_i(t)) \tag{7}$$

は，Klimontovich(-Dupree) 方程式の厳密解となる [12]。

$$\frac{\partial \hat{f}}{\partial t} + \boldsymbol{v} \cdot \nabla \hat{f} + \frac{q}{m}(\hat{\boldsymbol{E}} + \boldsymbol{v} \times \hat{\boldsymbol{B}}) \cdot \frac{\partial \hat{f}}{\partial \boldsymbol{v}} = 0 \tag{8}$$

ここで，$\hat{}$がついた量は，ミクロな量であることを表し，$\hat{\boldsymbol{E}}$, $\hat{\boldsymbol{B}}$ は，ミクロな電場と磁場である。これらは，ミクロなMaxwell方程式を満たす。ミクロなMaxwell方程式は，Maxwell方程式が線形方程式のため，通常の(マクロな)Maxwell方程式と同じ形である。

Klimontovich方程式は，個々の粒子に関する厳密な情報を持っており，扱いにくい。そこで，時間，空間の中で相互に影響を及ぼし合う粒子を適当なスケールで平均化する：

$$f(\boldsymbol{x}, \boldsymbol{v}, t) \equiv \langle \hat{f}(\boldsymbol{x}, \boldsymbol{v}, t) \rangle \tag{9}$$

左辺の量は，(アンサンブル) 平均化された量を表し，元の $\hat{f}$ に比べてマクロな量となっているので，$\hat{}$を落としてある。

つぎに，揺らぎを導入する。ミクロな量は，自分自身の平均量であるマクロな量と揺らぎからなると仮定する。ミクロな相空間密度，電場，磁場は，それぞれ次のように書き表される。

$$\begin{aligned}
\hat{f}(\boldsymbol{r}, \boldsymbol{v}, t) &= \langle \hat{f}(\boldsymbol{x}, \boldsymbol{v}, t) \rangle + \delta \hat{f}(\boldsymbol{x}, \boldsymbol{v}, t) \\
&= f(\boldsymbol{r}, \boldsymbol{v}, t) + \delta \hat{f}(\boldsymbol{r}, \boldsymbol{v}, t) && (10) \\
\hat{\boldsymbol{E}}(\boldsymbol{r}, \boldsymbol{v}, t) &= \boldsymbol{E}(\boldsymbol{r}, \boldsymbol{v}, t) + \delta \hat{\boldsymbol{E}}(\boldsymbol{r}, \boldsymbol{v}, t) && (11) \\
\hat{\boldsymbol{B}}(\boldsymbol{r}, \boldsymbol{v}, t) &= \boldsymbol{B}(\boldsymbol{r}, \boldsymbol{v}, t) + \delta \hat{\boldsymbol{B}}(\boldsymbol{r}, \boldsymbol{v}, t) && (12)
\end{aligned}$$

これらの関係式を，(8) 式に代入して平均化を行い揺らぎの1次を落とす (揺らぎの2次は残す) と，次の式が得られる。

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = -\frac{q}{m} \left\langle \left(\delta \hat{\boldsymbol{E}} + \boldsymbol{v} \times \delta \hat{\boldsymbol{B}} \right) \cdot \frac{\partial}{\partial \boldsymbol{v}} \delta \hat{f} \right\rangle \tag{13}$$

この右辺を具体的に評価した結果得られる式の例として，Fokker-Planck方程式が挙げられる：

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = \frac{\partial}{\partial \boldsymbol{v}} \cdot \left(\vec{\mathrm{D}} \cdot \frac{\partial f}{\partial \boldsymbol{v}} \right) \tag{14}$$

また，この式の右辺をゼロとし2体衝突を無視した式が，Vlasov方程式と呼ばれる式となる：

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f + \frac{q}{m}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0 \tag{15}$$

すなわち，Vlasov方程式はあくまで近似的に非粘性であることを覚えておいてほしい。ただし，プラズマの場合にはクーロン力による多体相互作用が場の量 $\boldsymbol{E}$ からもたらされるため，2体衝突を無視しても，多くの場合，非常によい近似を与えることを申し添える。

以上の通り，Klimontovich方程式はミクロな方程式であり，揺らぎを含んだミクロな量を代入し平均化を行いマクロな方程式とすることによって，(13) 式のような拡散項を導くことが可能となる。

5 点渦系での拡散

我々は，前章の事例を参考にし，点渦系にも同様の関係が成り立つと予想した。すなわち，点渦解をもつEuler方程式は，マクロな流体方程式としてのEuler方程式ではなく，Klimontovich方程式に対応するミクロな方程式であるとする。以後，点渦解を持つEuler方程式を,「ミクロなEuler方程式」とよぶ。プラズマの場合には，ミクロなKlimontovich方程式を平均化することにより，拡散項を含むFokker-Planck方程式や，近似的に非粘性のVlasov方程式が得られる。同様に，ミクロなEuler方程式からも，Fokker-Planck方程式やVlasov方程式に対応するマクロな方程式が得られるはずで，我々はVlasov方程式に対応する方程式が，(マクロな流体方程式としての)Euler方程式であると考える。一方，Fokker-Planck方程式に対応するマクロな方程式は知られてない。そこで，ここでは，前章と同様の手順により，Fokker-Planck方程式に対応する，拡散項を含んだマクロな方程式を導く。

出発点は，ミクロなEuler方程式である。ここで，マクロなEuler方程式と区別するために，変数に$\hat{}$を付した。

$$\frac{\partial}{\partial t}\hat{\omega}_z(\boldsymbol{r},t)+\hat{\boldsymbol{u}}(\boldsymbol{r},t)\cdot\nabla\hat{\omega}_z(\boldsymbol{r},t)=0 \tag{16}$$

ここで，$\hat{\boldsymbol{u}}(\boldsymbol{r},t)$は，ミクロな流れ関数$\hat{\psi}(\boldsymbol{r},t)$を用いて

$$\boldsymbol{u}(\boldsymbol{r},t)=-\hat{\boldsymbol{z}}\times\hat{\psi}(\boldsymbol{r},t) \tag{17}$$

と表されるミクロな速度場である。もちろん，ミクロな渦度は，デルタ関数により

$$\hat{\omega}_z(\boldsymbol{r},t)=\sum_i^N\Omega_i\delta(\boldsymbol{r}-\boldsymbol{r}_i(t)) \tag{18}$$

と表される。

先ほどと同様に，ミクロな量は，自分自身を平均化したマクロな量と揺らぎの和で表現されると仮定する。

$$\begin{aligned}
\omega_z(\boldsymbol{r},t) &\equiv \langle\hat{\omega}_z(\boldsymbol{r},t)\rangle=\left\langle\sum_i\Omega_i\delta(\boldsymbol{r}-\boldsymbol{r}_i)\right\rangle && (19)\\
\hat{\omega}_z(\boldsymbol{r},t) &= \langle\hat{\omega}_z(\boldsymbol{r},t)\rangle+\delta\hat{\omega}_z(\boldsymbol{r},t)\\
&= \omega_z(\boldsymbol{r},t)+\delta\hat{\omega}_z(\boldsymbol{r},t) && (20)\\
\hat{\boldsymbol{u}}(\boldsymbol{r},t) &= \boldsymbol{u}(\boldsymbol{r},t)+\delta\hat{\boldsymbol{u}}(\boldsymbol{r},t) && (21)
\end{aligned}$$

これを(16)式に代入し，平均化を行い1次の揺らぎの平均をゼロとすると，以下の式が得られる。

$$\frac{\partial}{\partial t}\omega_z(\boldsymbol{r},t)+\boldsymbol{u}(\boldsymbol{r},t)\cdot\nabla\omega_z(\boldsymbol{r},t)=-\nabla\cdot\langle\delta\boldsymbol{u}(\boldsymbol{r},t)\delta\omega_z(\boldsymbol{r},t)\rangle \tag{22}$$

左辺は全てマクロな量である。一方，右辺には揺らぎの積の平均が残ると考える。この項が，粒子性(ミクロ性)に由来する拡散項である。

$\delta\omega_z(\boldsymbol{r},t)$ の具体的表式を得るため，(16) 式に (20)，(21) 式で定義される揺らぎを含んだミクロな渦度を代入し，揺らぎに関して 1 次の項を集めた線形化方程式を用いる。

$$\frac{\partial}{\partial t}\delta\omega_z(\boldsymbol{r},t)+\boldsymbol{u}(\boldsymbol{r},t)\cdot\nabla\delta\omega_z(\boldsymbol{r},t)=-\delta\boldsymbol{u}(\boldsymbol{r},t)\cdot\nabla\omega_z(\boldsymbol{r},t) \tag{23}$$

この式の左辺第 2 項に現れる $\boldsymbol{u}(\boldsymbol{r},t)$，および右辺に現れる $\omega_z(\boldsymbol{r},t)$ は，マクロな量であることに注意する。マクロな量は，ミクロなスケールでは変化しない定数と見なすことができると考えると，上式は積分可能で，

$$\delta\omega_z(\boldsymbol{r},t)=\int_{-\infty}^{t}d\tau\delta\boldsymbol{u}\left(\boldsymbol{r}-(t-\tau)\boldsymbol{u},\tau\right)\cdot\nabla\omega_z(\boldsymbol{r},t) \tag{24}$$

を得る。ここで，$\delta\omega_z(\boldsymbol{r},t=-\infty)=0$ を仮定した。この結果を (22) 式右辺に代入すると，

$$\begin{aligned}-\nabla\cdot\langle\delta\boldsymbol{u}(\boldsymbol{r},t)\delta\omega_z(\boldsymbol{r},t)\rangle&=-\nabla\cdot(\vec{\vec{\eta}}\cdot\nabla\omega_z)\\ \vec{\vec{\eta}}&=\int_{-\infty}^{t}d\tau\langle\delta\boldsymbol{u}(\boldsymbol{r},t)\delta\boldsymbol{u}(\boldsymbol{r}-(t-\tau)\boldsymbol{u},\tau)\rangle\end{aligned} \tag{25}$$

となり，拡散係数に対応するテンソルの具体的表式が得られる。これが粒子性に由来する拡散項の表式である。よく知られた久保公式には時間相関しか含まれていないが，我々の結果にはマクロな流れに伴う位置のずれを考慮した相関が含まれていることが特徴となっている。

6 手軽にGPU

昨年，長らく使ってきた MDGRAPE-3 がとうとう入手不可能になったので，近年急速に注目されるようになった GPU がどの程度使えるのか検討した結果を掲載する。

通常，GPU で計算を行うためには，NVIDIA 社の GPU であれば，CUDA C と呼ばれる専用のプログラミング言語でプログラミングを行わなくてはならない。CUDA C は，C 言語に対する自然な拡張として実装されており，C 言語でのシミュレーション経験があれば，単にコンパイルできるプログラムを書くことは，さほど難しくない。しかし，コンパイルできることと GPU の性能を引き出せることは別次元の話であり，GPU 内部でのメモリアクセスの特性などを理解しなければ，真の性能を引き出すことは困難である。また，CUDA C はいまだ発展途上であり，GPU ハードウェアの発展に伴って頻繁に仕様が変更されるという問題点もある。よって，最新の API(Application Programming Interface) を追い続け自身のプログラムを更新し続けるのは，かなりの労力を必要とする。そこで，今回は，GPU 用のコードを C 言語プログラムから自動生成してくれる Goose とよばれるコンパイラを購入し [13]，点渦系のシミュレーションで必ず必要になる Biot-Savart 積分をどの程度高速化できるのか，試してみた。

Goose でコンパイルできるサンプルコードを，図 2 に示す。これは，i 番目の点渦の移動速度を求める Biot-Savart 積分 (のつもり) である。変数 num が粒子数を表す。この 2 重

```
#pragma goose parallel for precision ("double") loopcounter(i,j) result(u[i][0..1])
for (i=0; i<num; i++) {
     u[i][X] = 0.0;
     u[i][Y] = 0.0;
     for (j=0; j<num; j++) {
          ...
     }
}
```

図 2: Goose 用のサンプルコード。

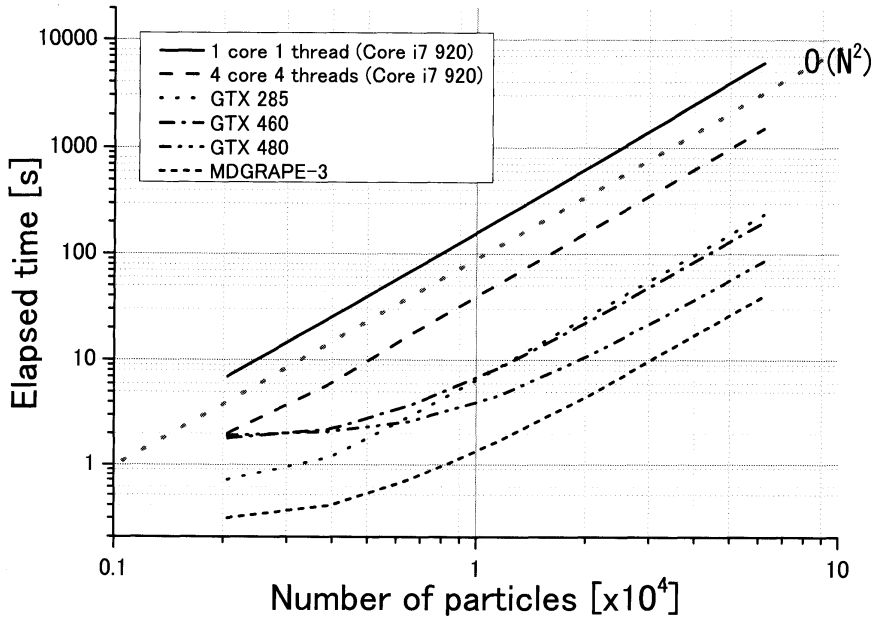

図 3: 粒子数をパラメタとした Biot-Savart 積分の計算時間の比較

ループの前に pragma 指示子により，Goose が必要としている情報を伝える。この指示子では，結果を求める際の計算精度指定を倍精度 (double)，ループカウンタが変数 i，j，結果を格納する配列が u[i][0]，u[i][1] である (サンプルコード中では，X = 0，Y = 1) ことを示している。なお，この pragma 指示子を理解できないコンパイラは無視するだけなので，このプログラムは通常の C コンパイラでもコンパイル可能である。Goose はこのような pragma 指示子に出会うと，続く 2 重ループを GPU で計算するコードに変換する。この変換先のコードは，CUDA C などのプログラミング経験が豊富な仙人らによって開発された成果物であり，Goose 開発者の川井氏によると，「黒魔術的に速い」そうである。

粒子数をパラメタとした Biot-Savart 積分の計算時間の比較結果を図 3，および表 1 に示す。GTX285 は 2009 年，GTX480 は 2010 年のハイエンド製品である。GTX460 は，GTX480 と同世代でミドルレンジ製品である。また，CPU を用いた 4 スレッドの計算は，インテルコンパイラの自動並列化を用いている。 Biot-Savart 積分は粒子数の 2 乗に比

表 1: MDGRAPE-3 の計算時間を 1 とした場合の計算時間の比率 (粒子数 $= 4 \times 10^4$ のケース)

環境	比率
CPU 1 コア 1 スレッド	151
CPU 4 コア 4 スレッド	38
GTX285	5.9
GTX460	5.0
GTX480	2.2
MDGRAPE-3	1.0

例する計算時間を必要とする。何粒子程度でこのスケーリングに合うようになるか分かりやすくするため，N^2 の傾きを図中に灰色の線で示してある。CPU を除いて，GPU，MDGRAPE-3 共に粒子数が少ない場合に計算速度が遅くなっているが，これは内部で並列化／ベクトル化する手間の方が，並列化／ベクトル化により高速化できる時間よりも長くなっていることを表している。よって，粒子数が多くなると，純粋に Biot-Savart 積分の計算にかかる時間が計算時間の大部分を占めるようになり，N^2 の傾きに漸近する。また，2009 年のハイエンド製品だった GTX285 は，多粒子側で 2010 年のミドルレンジ製品である GTX460 よりも計算速度が遅くなっており，1 年でこれだけ改善することに驚きを感じた。

最終的には，いまだに，ほぼ専用設計されている MDGRAPE-3 が速いが，GTX480 でも MDGRAPE-3 の 1/2 弱程度まで計算速度が肉薄してきている。MDGRAPE-3 が 100 万円程度のハードウェアである一方，GTX480 は当時 7 万円程度で購入しているので，コストパフォーマンスを考えると，GPU の時代になったと言えるかも知れない。また，pragma 指示子を書き加えてコンパイルし直すだけでこのような速度が出せる Goose の実力も十分実用に耐える，という印象を受けた。この結果を見て，研究室の学生には Goose を使ってもらおうと決意したのは，言うまでもない。

7 検討＆まとめ

今回の結果は，点渦数の多寡によって，対応するマクロな流体方程式が変わる可能性があることを示唆する。図 4 を見て欲しい。ミクロな Euler 方程式は，唯一無二の不変な式である。この式に対して揺らぎを考慮し平均化を行うと，(25) 式のように拡散項を含んだ結果が得られる訳だが，粒子数が少ない場合にはこの効果は小さいと考えられるため，結果的に拡散項はゼロとなると我々は考えている。

次に，シミュレーション結果に照らしあわせて検討してみよう。図 5 は，同符号点渦からなる渦塊二つが時間と共にマージする様子を点渦法で追跡したものである。$T = 0 \sim 0.5$

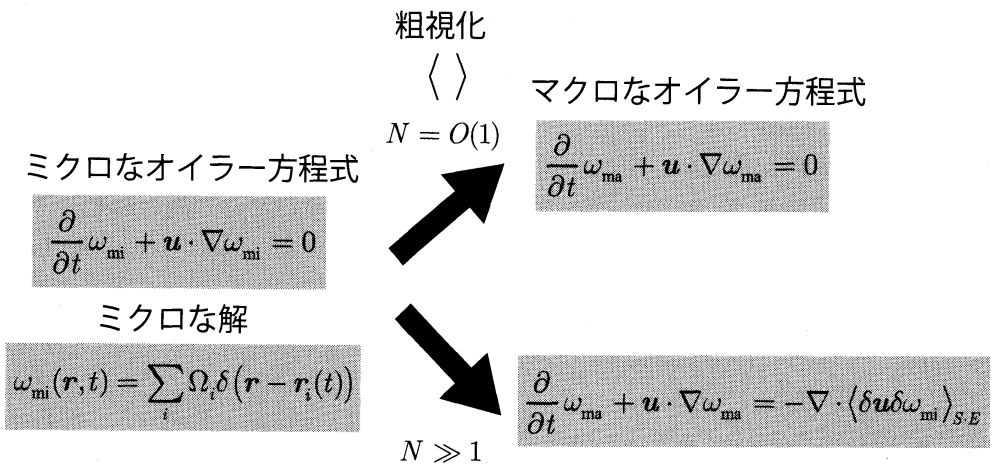

図 4: 点渦数の多寡により，マクロな式が非粘性になる場合もある。

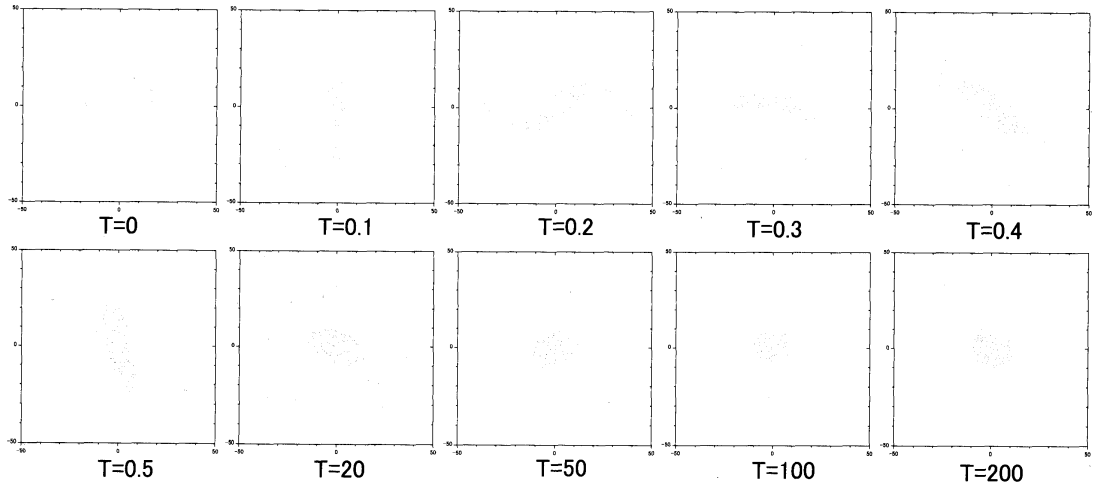

図 5: 同符号点渦のマージ。1 単位時間は，初期の渦塊一つの自転のタイムスケール程度である。粒子数 = 20056。$T = 0.5$ までは渦塊の外縁部にフィラメントが確認できる。$T = 20$ でフィラメントのアイデンティティは失われ始め，$T = 50$ では，もはや「雲」と呼ぶべき分布となっている。

にかけて，非常に高速にマージが進む。$T = 0.5$ 程度までの初期においては，マージに伴い大きな渦塊の周囲に生成されたフィラメントがくっきりと観察可能であるが，$T = 50$ 以降ではフィラメントとしてのアイデンティティはもはや失われ，「雲」と言ったほうが適切な分布となる。本来，点渦系はエネルギー保存系のため，そのトポロジーは保存される。すなわち，初期に生成されたフィラメントは，永遠にフィラメントであり続けるはずであるが，実際には図5が示す通り，フィラメントは拡散し，雲状分布になる。この原動力になっているのが，我々が求めた拡散項であろうと考えている。

また，我々が求めた拡散項は，エネルギー散逸につながるものではなく，渦度分布を文字通り拡散させるだけの効果しかもたない。拡散係数 ν が有限の Navier-Stokes 系の場合，運動エネルギー K の時間変化は，

$$\frac{dK}{dt} = -\nu \int |\omega_z(\boldsymbol{r}, t)| d\boldsymbol{r} \tag{26}$$

と与えられ，拡散係数 ν がエネルギー散逸につながるが，渦度方程式の場合には，有限の ν が渦度場のエネルギー散逸につながることはない (上記のような関係式は見あたらない) ため，エネルギー保存系であるという点渦系の前提を破ることもない。

まとめとして，点渦系に内在する拡散係数を解析的に求めた結果について検討した結果を報告した。これは，Green-Kubo 公式の拡張になっていると考えられ，また，点渦系が実は高 Reynolds 数の系に対するシミュレーション技法として有効であることを示唆している。

[1] P. K. Newton: *The N-Vortex Problem* (Springer-Verlag, Berlin, 2001) Chap. 1-3.

[2] 巽友正: 流体力学 (培風館, 東京, 1982) Chap. 9.

[3] L. Onsager: Nuovo Cimento Suppl. **6** (1949) 279.

[4] Y. Yatsuyanagi, Y. Kiwamoto, H. Tomita, M. M. Sano, T. Yoshida and T. Ebisuzaki: Phys. Rev. Lett. **94** (2005) 054502.

[5] G. Joyce and D. Montgomery: J. Plasma Phys. **10** (1973) 107.

[6] W. H. Matthaeus, W. T. Stribling, D. Martinez, S. Oughton and D. Montgomery: Physica D **51** (1991) 531.

[7] T. S. Lundgren and Y. B. Pointin: J. Stat. Phys. **17** (1977) 323.

[8] R. Robert and J. Sommeria: J. Fluid Mech. **229** (1991) 291.

[9] G. L. Eyink and K. R. Sreenivasan: Rev. Mod. Phys. **78** (2006) 87.

[10] A. Leonard: J. Comput. Phys. **37** (1980) 289.

[11] D. H. E. Dubin: Phys. Plasmas **10** (2003) 1338.

[12] Y. L. Klimontovich: *The statistical theory of non-equilibrium processes in a plasma* (MIT Press, Cambridge, Massachusetts, 1967).

[13] : http://www.kfcr.jp/.